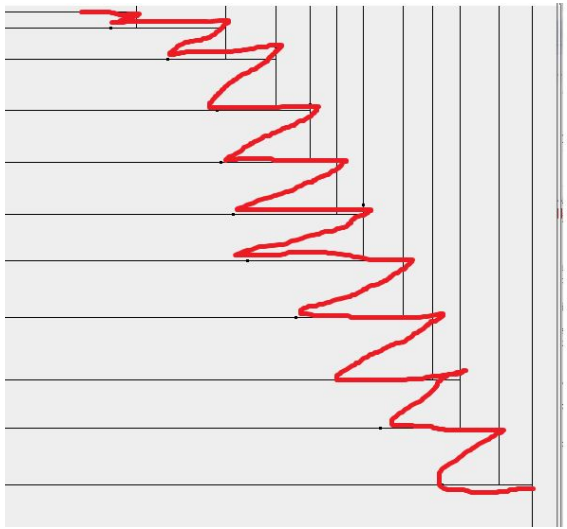
K-D Tree

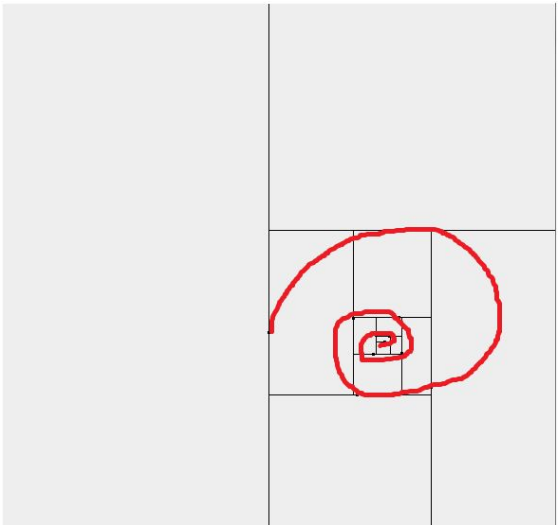
Andrew Fisher
Denver Blake
Uri Dvir
John Max Petrarca
Ivan Martinovic

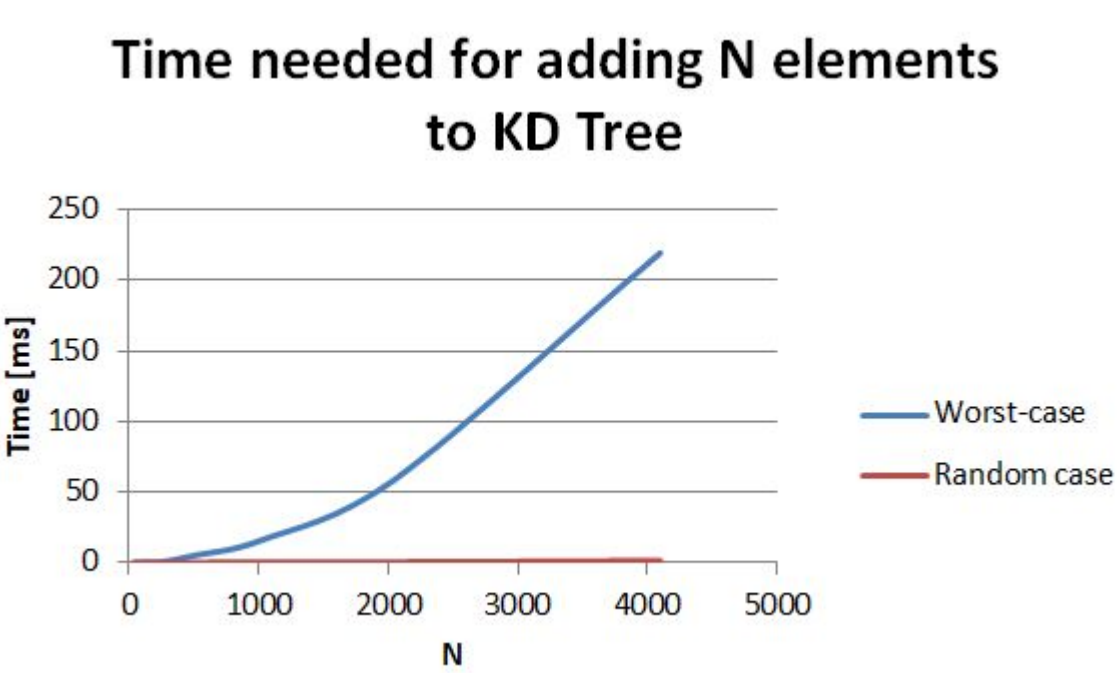
Introduction

- first invented in 1976 by Jon Louis Bentley and Michael Ian Shamos in their thesis: “Divide-and-Conquer in Multidimensional Space”
- a spatial tree structure used to efficiently perform nearest neighbor queries in k dimensions
- Each node contains its location and partitioning orientation
- Each node contains two children smaller and bigger referring to the child having a smaller or larger coordinate corresponding to the partitioning axis of the parent

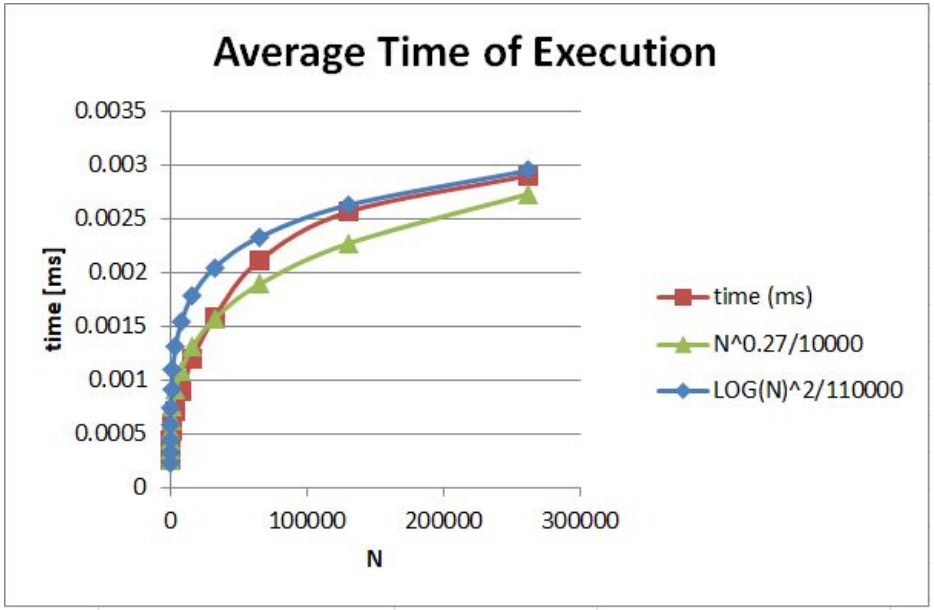
Worst-case Dataset for Insertion

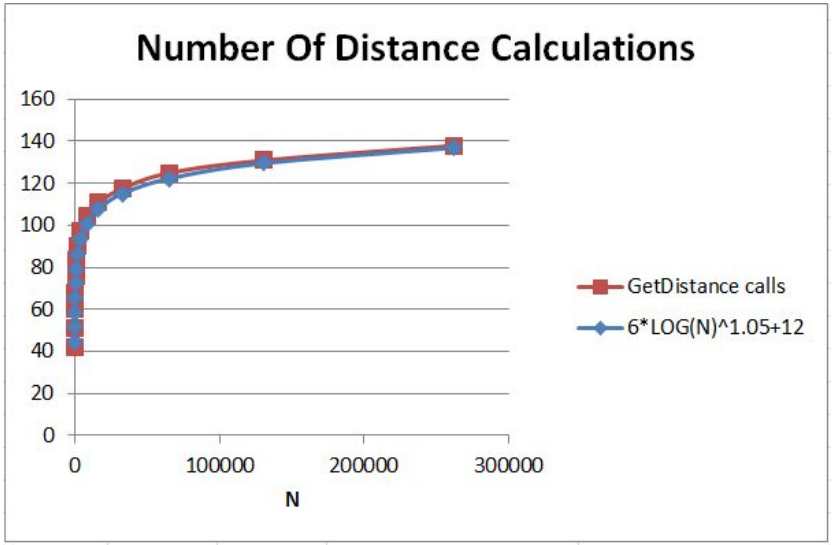





Time needed for adding N elements to KD Tree
Time [ms]
250
200
150
100
50
0
0
1000
2000
3000
4000
5000
N
Worst-case
Random case

Nearest Neighbor Performance Evaluation





Nearest Neighbor:K-D Tree vs Brute Force

N	KD Algorithm Time [ms]	Brute Force Algorithm Time [ms]
32	0.000252	0.000164
64	0.000266	0.000195
128	0.000319	0.000288
256	0.000376	0.000477
512	0.000443	0.000817
1024	0.000522	0.001706
2048	0.0006	0.003271
4096	0.000713	0.006362
8192	0.000891	0.012877
16384	0.001191	0.025486
32768	0.001588	0.051634
65536	0.002112	0.121298
131072	0.002565	0.252744
262144	0.002899	0.588444

Conclusion

- One of the first efficient data structures which allowed multidimensional data analysis and geometric optimization.
- The very first efficient data structure used for computing nearest and farthest neighbor queries in multi-dimensional space
- Can be extremely inefficient when unbalanced, both when adding elements and performing data analysis.
- These cases can occur when data has a sort of multi-dimensional zig-zag or spiral pattern